
climate*toolbox* Documentation

Release 0.1.4

ClimateImpactLab

Sep 18, 2019

Contents

1	climate_toolbox	3
1.1	Setting up this package	3
1.2	Features	5
1.3	Credits	5
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
6.1	0.1.4 (current version)	17
6.2	0.1.3 (2017-08-04)	17
6.3	0.1.2 (2017-07-25)	17
6.4	0.1.1 (2017-07-25)	17
6.5	0.1.0 (2017-07-24)	17
7	Indices and tables	19

Contents:

Tools for climate data wrangling

- Free software: MIT license
- Documentation: <https://climate-toolbox.readthedocs.io>.

1.1 Setting up this package

Justin Simcock: complete these instructions and then remove this section from the readme.

1. Create a matching repository on github if you haven't already:
 - a. go to <https://github.com/organizations/ClimateImpactLab/repositories/new>
 - b. enter the following information:
 - owner** ClimateImpactLab
 - Repository Name** climate_toolbox
 - Description** Tools for climate data wrangling
 - Privacy** Public
 - Do not initialize the repo with a readme, license, or gitignore!**
 - c. Press the big green button
2. Execute the following commands in this directory:

```
git init
git add .
git commit -m "initial commit"
git remote add origin git@github.com:ClimateImpactLab/climate_toolbox.git
git push -u origin master
```

3. Set up automated testing, coverage, updates, and docs:

a. Set up testing

- i. go to <https://travis-ci.org/> and sign in/sign up with github. Make sure travis has access to the repositories on your github account *and* those owned by the ClimateImpactLab by enabling access in your account settings.
- ii. Next to “My Repositories” click the ‘+’ icon
- iii. Flip the switch on ClimateImpactLab/climate_toolbox. If you don’t see it in the list, click ‘Sync account’, and make sure you are looking at the ClimateImpactLab repositories.
- iv. Push a change to your package (for example, you could delete this section). You should see tests start running on travis automatically.
- v. Start writing tests for your code in the `climate_toolbox/tests` folder. There are some examples already in there to get you started.

b. Set up docs

- i. go to <https://readthedocs.io> and sign in/sign up with github. Same deal - make sure readthedocs has access to your github account and ClimateImpactLab.
- ii. click ‘Import a project’, go to ClimateImpactLab, and refresh.
- iii. click the ‘+’ icon next to ClimateImpactLab/climate_toolbox. Pick a name for your project. The name has to be globally unique (not just within the ClimateImpactLab), so if you chose something with a common name you may have to rename the docs.
- iv. next time you push code to master, docs should build automatically. You can view them at https://climate_toolbox.readthedocs.io/en/latest/ (substitute whatever name you chose).

c. Set up updates

- i. go to <https://pyup.io>. you know the drill.
- ii. Go to your account, and click ‘+ Add Repo’. Sync.
- iii. Add ClimateImpactLab/climate_toolbox with “Dependency Updates” and “SafetyCI” turned on. Leave “Update Schedules” off.

d. Set up test coverage monitoring

- i. next up: <https://coveralls.io>. you got this.

e. Breathe easy. The next time you push code, these should all update for you! Now that wasn’t so bad, was it?

4. Develop with github:

- a. In general, it’s best to file an issue when you want to change something or when you’ve found a bug, then write tests which test a-priori assertions about desired behavior, and then write the minimum amount of code required to pass these tests. See the contributing docs for more workflow suggestions.

To run all tests on your local machine:


```
make test
```

5. Deploy

a. Set up deployment by registering the package on the python package index (PyPI):

i. Create an account on PyPI: <https://pypi.python.org/pypi>

ii. Register the package:

```
python setup.py register
```

iii. Encrypt & package your credentials so travis can deploy for you:

```
python travis_pypi_setup.py
```

b. When you're ready to deploy this package, make sure all your changes are committed. Then run:

```
bumpversion patch # (or minor or major)
git push
git push --tags
```

As soon as this new tagged commit passes tests, travis will deploy for you

c. Anyone (in the world) should now be able to install your package with

```
pip install [package-name]
```

1.2 Features

- TODO

1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install `climate_toolbox`, run this command in your terminal:

```
$ pip install climate_toolbox
```

This is the preferred method to install `climate_toolbox`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for `climate_toolbox` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/ClimateImpactLab/climate_toolbox
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/ClimateImpactLab/climate_toolbox/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use `climate_toolbox` in a project:

```
import climate_toolbox
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/ClimateImpactLab/climate_toolbox/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

climate_toolbox could always use more documentation, whether as part of the official climate_toolbox docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/ClimateImpactLab/climate_toolbox/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *climate_toolbox* for local development.

1. Fork the *climate_toolbox* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/climate_toolbox.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv climate_toolbox
$ cd climate_toolbox/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 climate_toolbox tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/ClimateImpactLab/climate_toolbox/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_climate_toolbox
```


This repository is a project of the [Climate Impact Lab](#)

5.1 Development Lead

- Justin Simcock <jsimcock@rhg.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.4 (current version)

- Support vectorized indexing for xarray ≥ 0.10 in `climate_toolbox.climate_toolbox._reindex_spatial_data_to_regions()` (GH #10)
- Support iteratively increasing bounding box in `_fill_holes_xr()` (GH #11).
- Support multiple interpolation methods (linear and cubic) in `_fill_holes_xr()` (GH #12).
- Fix bug causing tests to pass no matter what

6.2 0.1.3 (2017-08-04)

- Support passing a dataset (not just a filepath) into `load_baseline` and `load_bcsd` (GH #4)

6.3 0.1.2 (2017-07-25)

- merge in bug fixes

6.4 0.1.1 (2017-07-25)

- Various bug fixes (see GH #2)

6.5 0.1.0 (2017-07-24)

- First release on PyPI.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`